Eigendecomposition-Based Analysis of Video Images

C-Y. Chang, A. A. Maciejewski, and V. Balakrishnan

School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN 47907-1285

ABSTRACT

We present a fast algorithm for computing the singular value decomposition (SVD) of a matrix consisting of the frames from a video sequence. The computational efficiency of this algorithm derives from the observation that portions of a video sequence will consist of sets of correlated frames. We then show that the information obtained from the SVD can be used to analyze video sequences to obtain information such as scene breaks, scene query, reduced-order shot representation and key frame determination. We illustrate this approach on several video sequences.

Keywords: Eigenimages, eigendecomposition, singular value decomposition (SVD), video analysis

1. INTRODUCTION

Image and video analyses inherently require the handling of a large volume of information. Thus efficient mechanisms for the organization and classification of data, as well as for data query and retrieval, are essential in managing video and image databases. Our objective in this article is to explore the use of reduced order representations towards addressing some of the issues that arise in video database management.

The specific reduced order representation that we will consider is based on eigenspace methods. Variously referred to as principal component analysis methods and Karhunen-Loeve transformation methods, eigenspace methods have been used extensively in a variety of applications such as face characterization and recognition, lip-reading, object recognition, pose detection, visual tracking and inspection, and video analysis. All of these applications are based on taking advantage of the fact that a set of highly correlated images can be approximately represented by a small set of eigenimages, which is also the case with several video applications.

The main drawback in the application of eigenspace methods for analyzing video sequences is the excessive amount of computation required to generate the eigenimages. One way of dealing with this issue is to compress the image information in the time and spatial domains. However, the degree of compression is limited by accuracy requirements. In this work, we present a different technique (that can be used instead of or along with compression techniques) for reducing the computation involved in the eigenspace approach for video analysis. An additional advantage of our approach is that it provides the right singular vectors in the singular value decomposition as well. We demonstrate how this additional information can be used for scene break detection and key frame identification in video analysis.

The organization of the paper is as follows. In Section 2, we present the mathematical preliminaries underlying our algorithm and in Section 3 the algorithm itself. We then illustrate the application of our techniques on several video sequences, considering specifically the problems of scene break detection, shot representation, key frame identification and frame query.

2. PRELIMINARIES

A frame is an $h \times v$ array of square pixels with intensity values normalized between 0 and 1. Thus, a frame will be represented by a matrix $X \in [0, 1]^{h \times v}$. Since we will be considering sets of related frames, it will be convenient to represent a frame equivalently as a vector, obtained simply by “row-scanning”, i.e., concatenating the rows to obtain the frame vector $x$ of length $m = hv$:

$$x = \text{vec}(X^T).$$

(1)

The frame data matrix of a set of frames $X_1, \ldots, X_n$ is an $m \times n$ matrix, denoted $X$, and defined as

$$X = [x_1 \cdots x_n],$$

Further author information, contact Professor A. A. Maciejewski, Phone: (765) 494-9855, Fax: (765) 494-6951, E-mail: maciejw@ecn.purdue.edu
with typically $m \gg n$.

The singular value decomposition (SVD) of $X$ is given by

$$X = U\Sigma V^T,$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal, and $\Sigma \in \mathbb{R}^{m \times n}$, with $\Sigma^T = [\Sigma_0 \mathbf{0}]^T$, where $\Sigma_0 = \text{diag}(\sigma_1, \ldots, \sigma_n)$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$, and $\mathbf{0}$ is an $n$ by $m - n$ zero matrix. (When the singular values $\sigma_i$ are not ordered, we will refer to the decomposition as an “unordered” SVD.) The SVD of $X$ plays a central role in several important imaging applications such as image compression, pattern recognition and pose detection.

The columns of $U$, denoted $u_i$, $i = 1, \ldots, m$, are referred to as the eigenimages of $X$; these provide an orthonormal basis for the frames within the video sequence, ordered in terms of importance. Thus, the first eigenimage can be interpreted as a frame containing the “most common” information from all of the frames of the video sequence. Likewise, the first $k$ eigenimages provide a basis for the best $k$-dimensional representation of the frames in the video sequence. Thus eigenimages provide a natural, ordered hierarchy for the compressed representation of information to within a user-defined level of accuracy. In addition to the eigenimages, the information contained in the vectors comprising $V$ (known as the right singular vectors) can be used in video analysis. Specifically, the components of the $i$th column of $V$ measure how much each individual frame contributes to the $i$th eigenimage.

We will demonstrate the use of eigenspace methods by using the frame query problem of video analysis as an example. Given a test frame, the objective is to find the “closest” frame from a video sequence. When the distance between frames is measured via the Euclidean norm of the difference between the corresponding frame vectors, the SVD provides a framework for trading off computational effort and accuracy. With $x_{\text{test}}$ denoting the frame vector of the test frame, the objective is to find $i$ that minimizes $||x_i - x_{\text{test}}||$, i.e., the frame that best matches $x_{\text{test}}$. A direct implementation of this minimization is prohibitively expensive, due to the typically large values of $m$ and $n$.

A standard technique for reducing computation is to perform the comparisons using the eigenimages. Representing $x_{\text{test}} - x_i$ in terms of the eigenimages yields

$$x_{\text{test}} - x_i = \sum_{j=1}^{k} (x_{\text{test}} - x_i)^T u_j u_j + \sum_{j=k+1}^{n} (x_{\text{test}} - x_i)^T u_j u_j + \sum_{j=n+1}^{m} (x_{\text{test}} - x_i)^T u_j u_j.$$ 

The first term in the above equation represents the projection of $x_{\text{test}} - x_i$ on the $k$-dimensional subspace spanned by $\{u_1, \ldots, u_k\}$; thus, it can be regarded as a “$k$-dimensional approximation” to $x_{\text{test}} - x_i$. The second term is the projection of $x_{\text{test}} - x_i$ on to the orthogonal complement of the range of the shades that comprise $X$; thus, if $x_{\text{test}}$ were one of the original frames, this term represents the error in its $k$-dimensional approximation. The third term represents the component of $x_{\text{test}}$ that lies outside the range of $X$. If $x_{\text{test}}$ is generated by the same statistics as the frame vectors of the frame data matrix, then it can be shown that the first term yields the best “$k$-dimensional” approximation to $x_{\text{test}} - x_i$, over all possible $k$-dimensional subspaces. Thus,

$$||x_{\text{test}} - x_i||^2 \approx \sum_{j=1}^{k} (x_{\text{test}}^T u_j - x_i^T u_j)^2,$$

and

$$\min_{i=1, \ldots, n} ||x_{\text{test}} - x_i||^2 \approx \min_{i=1, \ldots, n} \sum_{j=1}^{k} (x_{\text{test}}^T u_j - x_i^T u_j)^2.$$ 

Note that $x_i^T u_j$ can be precomputed. Calculating the right-hand side of (6) requires performing $k$ times an $m$-dimensional inner product, followed by the norm computation of the difference between two $k$-dimensional vectors, performed $n$ times; this takes $O(km) + O(kn)$ operations. Calculating the left-hand side directly requires finding the norm of the difference between two $m$-dimensional vectors, $n$ times; this takes $O(nm)$ operations. Typically, $k \ll m$ and $k \ll n$, and thus in practice, the left-hand side of (6) is computed only approximately, by evaluating the right-hand side.

In practice, the singular vectors $u_i$ are not known or computed exactly, and instead estimates $q_1, \ldots, q_k$ which form a $k$-dimensional basis are used. The accuracy of these estimates then depends on three factors: the properties
of $X$, the dimension $k$, and the quality of the estimates $q_k$. The measure we will use for quantifying this accuracy is the “energy recovery ratio”, denoted $\rho$, and defined as
\[
\rho(X, q_1, \ldots, q_k) = \frac{\sum_{i=1}^{k} \|q_i^T X\|^2}{\|X\|^2_F},
\]
where $\| \cdot \|_F$ denotes the Frobenius norm. Note that if the $q_k$s are orthonormal, $\rho \leq 1$, and for any given $k$ achieves a maximum value of $(\sum_{i=1}^{k} \sigma_i^2)/(\sum_{i=1}^{n} \sigma_i^2)$ when span($q_1, \ldots, q_k$) = span($u_1, \ldots, u_k$).

The principal calculation required in the above approach is the precomputation of estimates of the singular vectors $u_1, \ldots, u_k$ of the $m \times n$ matrix $X$. This is a very computationally expensive operation when $m$ and $n$ are very large. Reducing this computational expense by exploiting any correlation between frame vectors has been the subject of previous works. One class of techniques relies on updating a small set of eigenimages by recursively adding one frame at a time. In Murakami et al.,11 the number of eigenimages is fixed, while in Chandrasekaran et al.12 this number is adjusted based on the content of the added frame. Another approach is to efficiently compute $X^T X$ and its eigendecomposition, from which the SVD of $X$ is constructed.13 We address the problem of computing the SVD of $X$ using a fundamentally different approach that is considerably faster than these methods when the frame vectors are “correlated”, as in video sequences. We describe the algorithm in the next section.

### 3. A Fast Eigendecomposition Algorithm

Our algorithm relies on the observation that the right singular vectors from the SVD of a frame data matrix $X$ from a single correlated video shot (without scene breaks) are very close to perfect harmonics with fundamental frequency $2\pi/n$. Moreover, the ordered right singular vectors correspond to harmonics of increasing frequency. (See Chang et al.14 for details.) If this were exactly true, then a real eigendecomposition of $X^T X$ would be given by
\[
X^T X = H D H^T, \tag{7}
\]
where $D$ equals the $n \times n$ matrix
\[
\text{diag}\left( P(\omega^0), P(\omega^1), P(\omega^2), \cdots \right), \tag{8}
\]
and $H$ consists of the first $n$ columns of
\[
\sqrt{\frac{2}{n}} \left[ \frac{1}{\sqrt{2}} f_1 \begin{array}{c} \mathbb{R} f_2 \mathbb{R} f_3 \mathbb{R} f_3 \cdots \end{array} \right] = \begin{bmatrix} \frac{1}{\sqrt{2}} & c_0 & -s_0 & c_0 & -s_0 & \cdots \\ \frac{1}{\sqrt{2}} & c_1 & -s_1 & c_2 & -s_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \frac{1}{\sqrt{2}} & c_{n-1} & -s_{n-1} & c_2(n-1) & -s_2(n-1) & \cdots \end{bmatrix} \tag{9}
\]

where $c_k = \cos(k\theta)$ and $s_k = \sin(k\theta)$.

Our objective is to determine the first $k$ left singular vectors of $X$. Let $p$ be such that the harmonic content of the first $k$ singular vectors are essentially restricted to the band $[0, 2\pi p/n]$ ($p$ is typically not much larger than $k$). Let $H_p$ denote the matrix comprising the first $p$ columns of $H$ (i.e., the first $p$ columns of the matrix given in (9)). Then the first $k$ singular values $\sigma_1, \ldots, \sigma_k$ and the corresponding left singular vectors $\tilde{u}_1, \ldots, \tilde{u}_k$ of $XH_p$ serve as excellent estimates to those of $X$. (Note that $XH_p$ typically has far fewer columns than $X$, so that its SVD can be computed much more quickly.) Moreover, the accuracy of the approximated singular vectors with spectra concentrated around “lower” frequencies will tend to be better, i.e., the smaller $i$ is, the better estimate $\tilde{u}_i$ is of $u_i$.

Our ultimate goal is to guarantee, upon termination, that the energy recovery ratio $\rho(X, \tilde{u}_1, \ldots, \tilde{u}_k)$ exceeds a user-specified threshold $\mu$. However, note that $\rho(X, \tilde{u}_1, \ldots, \tilde{u}_k)$ depends critically on $k$ and $\tilde{u}_1, \ldots, \tilde{u}_k$, neither of which are available a priori. We address this issue using the fact that
\[
\rho(X, \tilde{u}_1, \ldots, \tilde{u}_p) \geq \rho(X^T, h_1, \ldots, h_p), \tag{10}
\]
where $\mathbf{h}_i$ denotes the $i$th column of $H$. The right-hand side of (10) is readily computed; and ensuring that $\rho(X^T, h_1, \ldots, h_p) \geq \mu$ in turn guarantees that $\rho(X, \tilde{u}_1, \ldots, \tilde{u}_p) \geq \mu$. 


It turns out that \( \rho(X^T, h_1, \ldots, h_p) \) is a very conservative lower bound for \( \rho(X, u_1, \ldots, u_p) \), with the quality of the bound improving uniformly with increasing \( p \). For fixed \( p \), \( \rho(X, u_1, \ldots, u_k) \) behaves as a very good lower bound to \( \rho(X, u_1, \ldots, u_k) \) for small \( k \), and is very well approximated from below by \( \rho(X^T, h_1, \ldots, h_k) \) for large \( k \).

In summary, when \( p \) is chosen so as to satisfy \( \rho(X^T, h_1, \ldots, h_p) \geq \mu \), the quantity \( \rho(X, u_1, \ldots, u_k) \) turns out to exceed \( \mu \) for some \( k \leq p \), with \( u_1, \ldots, u_k \) being very good estimates for \( u_1, \ldots, u_k \); and \( \hat{\sigma}_1, \ldots, \hat{\sigma}_k \) are very good estimates for \( \sigma_1, \ldots, \sigma_k \). The energy recovery ratio \( \rho(X, u_1, \ldots, u_k) \) can be efficiently approximated by \( \sum_{i=1}^k \hat{\sigma}_i^2 / ||X||_F^2 \).

The entire algorithm can be summarized as follows.

1. Multiply \( X \) by \( H \). This is performed computationally efficiently as follows:
   a. Compute the product \( Y = \sqrt{n} X F \) by forming a matrix whose \( i \)th row is the FFT of the \( i \)th row of \( X \).
   b. Construct \( Z = \sqrt{\frac{F}{n}} X H \) as the first \( n \) columns of the matrix
      \[ \frac{1}{\sqrt{F}} y_1 \bar{y}_2 \bar{y}_3 \bar{y}_4 \cdots \]
      where \( y_i \) denotes the \( i \)th column of \( Y \).
2. Determine the smallest number \( p \) such that \( \rho(X^T, h_1, \ldots, h_p) > \mu \), where \( \mu \) is the user-specified reconstruction ratio.
3. Compute the SVD of the matrix comprising the first \( p \) columns of \( Z \).
4. Return \( u_1, \ldots, u_k \) such that \( \sum_{i=1}^k \hat{\sigma}_i^2 / ||X||_F^2 > \mu \).

We briefly analyze the computational expense of our algorithm. The cost incurred in Step 1, i.e., performing the FFT of each row of \( X \), requires \( O(mn \log_2 n) \) flops. Step 2, that of estimating \( p \), requires \( O(mp) \) flops. Finally, the cost of computing the SVD of the matrix comprising the first \( p \) columns of \( \sqrt{\frac{F}{n}} X H \) is of order \( O(mp^2) \). If \( p \ll n \), then the total computation required is approximately \( O(mn \log_2 n) \). This compares very favorably with the direct SVD approach which requires \( O(mn^2) \) flops, and in most cases with updating SVD method as well, which requires \( O(mnk^2) \) flops.

4. VIDEO ANALYSIS USING THE RIGHT SINGULAR VECTORS

We now illustrate how the information from the right singular vectors can be used for various video analysis applications such as detection of scene breaks, key frame identification and frame query.

4.1. Detecting scene breaks

In a set of highly correlated frames such as those that arise within a single shot, it turns out that all of the frames contribute “equally” to the eigenimages. In other words, the components of the first few right singular vectors (i.e., the columns of \( V \) in (3)) are slowly varying functions of the frame index. In fact, for certain special cases, it can be established mathematically that the right singular vectors consist of pure harmonics, in increasing order of frequency. Thus, for a video sequence consisting of correlated frames (such as those within a single shot), the elements of the right singular vectors will be smooth functions of their index. Therefore, any abrupt changes between successive elements in the right singular vectors can be used as indicators of scene breaks.

Let \( k \) be the dimension of the reduced-order approximation of the frames in the video sequence (in our experiments we have used \( k = 6 \)). Let \( \Delta v_{i, \text{max}} \) denote the the absolute value of the largest discontinuity between successive elements of the right singular vector \( v_i \). Then, with \( u(\cdot) \) being the unit step function and \( \lambda \) being a threshold parameter, we define the scene break measure \( \beta_j \) at frame \( j \) as

\[
\beta_j = \sum_{i=1}^k u \left( \frac{|v_{i,j} - v_{i,j-1}| - \lambda \Delta v_{i, \text{max}}}{\Delta v_{i, \text{max}}} \right)
\]  

(11)

At frame index \( j \), any “discontinuity” in the \( i \)th singular vector that exceeds \( \lambda \Delta v_{i, \text{max}} \) contributes a vote for a scene break at frame \( j \); these votes are added across all singular vectors, without any weighting. (It is also possible to replace the unit step function in the definition of \( \beta_j \) by other monotonic smooth functions so as to obtain “continuous” scene break measures.) If \( \beta_j \) exceeds a user-defined threshold \( \eta \), the algorithm reports a scene break at frame \( j \).
Figure 1 illustrates these ideas on a simple example of a video sequence of a table tennis game that contains one scene break. The first shot, contained in frames 1 through 66, starts by focusing on the ball, and then zooming out to show the first player serving. The second shot, contained in frames 67 through 96, shows the second player returning the serve. The first and the last frames of each of these shots is shown in Figure 1. In addition, two of the right singular vectors from the singular value decomposition of the complete video sequence are shown to illustrate how scene breaks are reflected in the characteristics of the singular vectors. It can be observed that both singular vectors exhibit a significant change in behavior between frames 66 and 67, thus indicating a scene break. (Our algorithm detected this as the only scene break for a range of values for the threshold parameter 0.26 ≤ λ ≤ 1. Moreover, v2 is dominated by the frames from the second shot; this is verified by inspecting the corresponding eigenimage. Similar comments can be made about v3 as well.

A second example with many more scene breaks, consisting of 190 frames of a “Wallace & Gromit” cartoon, is presented in Figure 2. In part (a) of this figure, a plot of the scene break measure is shown as a function of the frame index. The scene breaks detected using this measure, with λ = 40% and η = 1.2, were used to divide the video sequence into a set of (correlated) shots. Part (b) shows the first and last frames of every shot thus detected.

5. VIDEO SEQUENCE REPRESENTATION AND FRAME QUERY

Once scene breaks have been identified, a shot will consist of a sequence of highly correlated frames. In this case, the eigenimages can be computed very efficiently, and the optimal lower dimensional subspace representation can be determined. This representation can be used to answer several questions that arise in shot representation. For example, assume that a user of the video database has a query, and wishes to identify all shots where this frame occurs, as well as the frame index within that shot. The shot identification can be performed quickly by projecting the given frame on to the eigenimages from all shots in the database that are likely candidates, and selecting the best match. Once the shot has been identified, the frame index can be determined by searching for that frame from the shot whose projections on the individual eigenimages match best with those of the given frame. The lower dimensional representation can also be used to identify the single frame from within the shot that best represents the shot; this is simply the frame that contributes the most to the dominant eigenimages.

Some of the above ideas are illustrated in Figures 3 and 4. Figure 3 displays a video sequence consisting of a single shot of 82 frames of a hockey game, where the camera pans to center on the player with the puck. Three of the dominant eigenimages are shown, along with the location of the frames in the video sequence projected on to the corresponding three-dimensional eigen-subspace. The location of a given frame within the video sequence can be quickly determined by projecting the frame onto the three-dimensional eigen-subspace, and comparing the resulting three numbers with those stored for each frame in the sequence. The eigenspace decomposition can also be used to automatically identify an optimal key frame for representing a video sequence. One approach is to simply choose the frame which contributes the most to the dominant eigenimage, i.e., the index of the component of v1 with the largest magnitude. These ideas are illustrated in Figure 3.

Figure 4 shows the automatic key frame identification scheme applied to the cartoon sequence in Figure 2. In particular, the key frames corresponding to each separate shot in the sequence are shown. Note that this scheme can also be used to identify correlation between shots that are separated by other shots. For example, shots 3, 5, 7 and 12 are all shots of a flying duck; therefore the frames that comprise these shots are clustered in eigenspace. This is illustrated in part (b) of Figure 4, where the frames are shown clustered in the principal two-dimensional eigenpace. Similarly, shots 6 and 5, and 11 and 13 have similar content, and are seen clustered as well.

6. CONCLUSION

For video sequences consisting of sequences of correlated frames, we have presented a technique for efficiently computing their reduced-order representation, that is, the complete truncated SVD of the matrix consisting of the frames from a video sequence. We then demonstrate how the right singular vectors from the truncated SVD can be used to analyze video sequences to obtain information such as scene breaks, scene query, reduced-order shot representation and key frame determination.
REFERENCES
15. This 190 frame video sequence is called “Capturing Feathers McGraw (the penguin)” from “The Wrong Trousers”, one of a sequence of “Wallace & Gromit” animation short movies by Nick Park. This sequence is available in MPEG format from the web site:
http://wwwzenger.informatik.tu-muenchen.de/persons/paula/mpeg/index.html
**Figure 1.** An illustration of how eigenspace decomposition can be used to identify scene breaks.
Figure 2. Part (a) shows the beginning and ending frames for each separate shot as determined by applying the scene break measure to the entire video sequence. Part (b) shows the scene break measure as a function of the frame index; sharp peaks here suggest possible scene breaks. Note that the algorithm returns a scene break between frames 80 and 81. While this may not appear from the still images presented, it turns out that the shot consisting of frames 67 through 80 is a rapid camera pan, while the shot consisting of frames 81 through 85 has the camera stationary.
Figure 3. An example of a video shot, its first three eigenimages, and the use of eigenspace methods for scene representation, key frame identification and frame query.
Figure 4. Part (a) shows the key frames identified for each of the 14 shots from the cartoon sequence given in Figure 2. Part (b) illustrates that shots with similar content appear clustered in any eigenspace, with the principal two-dimensional eigenspace shown here. Shots with content not similar to that of any other shot have been omitted for clarity.